

Package ‘HDMD’

July 2, 2014

Type Package

Title Statistical Analysis Tools for High Dimension Molecular Data (HDMD)

Version 1.2

Date 2013-2-26

Author Lisa McFerrin

Maintainer Lisa McFerrin <lgmcferr@ncsu.edu>

Depends psych, MASS

Suggests scatterplot3d

Description High Dimensional Molecular Data (HDMD) typically have many more variables or dimensions than observations or replicates ($D \gg N$). This can cause many statistical procedures to fail, become intractable, or produce misleading results. This package provides several tools to reduce dimensionality and analyze biological data for meaningful interpretation of results. Factor Analysis (FA), Principal Components Analysis (PCA) and Discriminant Analysis (DA) are frequently used multivariate techniques. However, PCA methods `prcomp` and `princomp` do not reflect the proportion of total variation of each principal component. `Loadings.variation` displays the relative and cumulative contribution of variation for each component by accounting for all variability in data. When $D \gg N$, the maximum likelihood method cannot be applied in FA and the the principal axes method must be used instead, as in `factor.pa` of the `psych` package. The `factor.pa.ginv` function in this package further allows for a singular covariance matrix by applying a general inverse method to estimate factor scores. Moreover, `factor.pa.ginv` removes and warns of any variables that are constant, which would otherwise create an invalid covariance matrix. `Promax.only` further allows users to define rotation parameters during factor estimation. Similar to the Euclidean distance, the Mahalanobis distance estimates the relationship among groups. `pairwise.mahalanobis` computes all such pairwise Mahalanobis distances among groups and is useful

for quantifying the separation of groups in DA. Genetic sequences are composed of discrete alphabetic characters, which makes estimates of variability difficult. `MolecularEntropy` and `MolecularMI` calculate the entropy and mutual information to estimate variability and covariability, respectively, of DNA or Amino Acid sequences. Functional grouping of amino acids (Atchley et al 1999) is also available for entropy and mutual information estimation. Mutual information values can be normalized by NMI to account for the background distribution arising from the stochastic pairing of independent, random sites. Alternatively, discrete alphabetic sequences can be transformed into biologically informative metrics to be used in various multivariate procedures. `FactorTransform` converts amino acid sequences using the amino acid indices determined by Atchley et al 2005.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-02-27 07:31:03

R topics documented:

HDMD-package	3
AA54	4
AAMetric	5
AAMetric.Atchley	6
AminoAcids	7
bHLH288	8
factor.pa.ginv	9
FactorTransform	11
Loadings.variation	13
MolecularEntropy	14
MolecularMI	15
NMI	16
pairwise.mahalanobis	18
Promax.only	19

Index **21**

Description

High Dimensional Molecular Data (HDMD) typically have many more variables or dimensions than observations or replicates ($D \gg N$). This can cause many statistical procedures to fail, become intractable, or produce misleading results. This package provides several tools covering Factor Analysis (FA), Principal Components Analysis (PCA) and Discriminant Analysis (DA) to reduce dimensionality and analyze biological data for meaningful interpretation of results. Since genetic (DNA or Amino Acid) sequences are composed of discrete alphabetic characters, entropy and mutual information are often used to estimate variability and covariability, respectively. Alternatively, discrete alphabetic sequences can be transformed into biologically informative metrics to be used in various multivariate procedures. This package provide molecuclr entropy and mutual information estimates as well as a metric transformation to convert amino acid letters into indices determined by Atchley et al 2005.

Details

Package:	HDMD
Type:	Package
Version:	1.1
Date:	2012-9-19
License:	GPL (>=2)
LazyLoad:	yes

Author(s)

Lisa McFerrin Maintainer: Lisa McFerrin <lgmferr@ncsu.edu>

References

Atchley, W.R., Zhao, J., Fernandes, A. and Druke, T. (2005) Solving the sequence "metric" problem: Proc. Natl. Acad. Sci. USA 102: 6395-6400

Atchley, W.R. and Fernandes, A. (2005) Sequence signatures an the probabilistic identification of proteins in the Myc-Max-Mad network. Proc. Natl. Acad. Sci. USA 102: 6401-6406

Revelle, W. (in preparation) An Introduction to Psychometric Theory with applications in R. Springer at <http://personality-project.org/r/book>

See Also

[psych](#) ~~

Examples

```

data(AA54)
#perform Factor Analysis on HDMD where D>>N
Factor54 = factor.pa.ginv(AA54, nfactors = 5, m=3, prerotate=TRUE, rotate="Promax", scores="regression")
Factor54

data(bHLH288)
bHLH_Seq = as.vector(bHLH288[,2])
grouping = t(bHLH288[,1])

#Transform Amino Acid Data into a biologically meaningful metric
AA54_MetricFactor1 = FactorTransform(bHLH_Seq, Replace=AAMetric, Factor=1, alignment=TRUE)

#Calculate the pairwise mahalanobis distances among groups given a discriminant function
AA54_lda_Metric1 = lda(AA54_MetricFactor1, grouping)
AA54_lda_RawMetric1 = as.matrix(AA54_MetricFactor1)
AA54_lda_RawMetric1Centered = scale(AA54_lda_RawMetric1, center = TRUE, scale = FALSE)
AA54_lda_RawMetric1Centered[c(20:25, 137:147, 190:196, 220:229, 264:273),1:8]
plot(-1*AA54_lda_RawMetric1Centered[,1], -1*AA54_lda_RawMetric1Centered[,2], pch = grouping, xlab="Canonical Var
lines(c(0,0), c(-15,15), lty="dashed")
lines(c(-35,25), c(0,0), lty="dashed")

Mahala_1 = pairwise.mahalanobis(AA54_lda_RawMetric1Centered, grouping)
D = sqrt(Mahala_1$distance)
D

```

AA54

*Normalized Amino Acid Indices quantifying 54 various attributes***Description**

From approximately 500 indices listed in www.genome.jp/aaindex as of 2005, a sample of 54 Amino Acid Indices were selected to represent the range of structural and functional attributes. Each index was normalized to have mean 0 and variation 1.

Format

AA54 is a matrix of 54 indices (columns) quantifying attributes for the 20 amino acids (rows). Amino acids are represented by their single character abbreviation and sorted alphabetically. Indices are normalized to have mean = 0 and variation = 1.

Source

www.genome.jp/aaindex

Examples

```

data(AA54)
AA54

```

Description

Atchley et al 2005 performed factor analysis on a set of Amino Acid Indices (AA54) and inferred a 5 factor latent variable structure relating amino acid characteristics using SAS. An equivalent analysis was performed using `factor.pa.ginv` from the HDMD package in R. Based on the relationship between factors and variable descriptions, the latent variables are defined as Factor1 (PAH): Polarity, Accessibility, Hydrophobicity; Factor2 (PSS): Propensity for Secondary Structure; Factor3 (MS) : Molecular Size; Factor4 (CC): Codon Composition; Factor5 (EC): Electrostatic Charge. While the Factor Analysis loadings were the same, R and SAS calculated scores slightly differently. AAMetric are scores from the R factor analysis which convey the similarities and differences among amino acids (rows) for each latent variable (columns).

Format

Rows are alphabetized Amino Acids and the 5 columns are factors where Factor1 (PAH): Polarity, Accessibility, Hydrophobicity; Factor2 (PSS): Propensity for Secondary Structure; Factor3 (MS) : Molecular Size; Factor4 (CC): Codon Composition; Factor5 (EC): Electrostatic Charge.

Details

54 Amino Acid Indices were selected from www.genome.jp/aaindex to quantify Amino Acid Similarities. Using Factor Analysis on 5 factors, interpretable latent variables were determined to quantify Amino Acid attributes. These are the scores from factor analysis calculated by `factor.pa.ginv` in R.

Source

Method similar to Atchley, W. R., Zhao, J., Fernandes, A. and Drueke, T. 2005. Solving the sequence "metric" problem: Proc. Natl. Acad. Sci. USA 102: 6395-6400.

See Also

[AAMetric.Atchley](#), [factor.pa.ginv](#)

Examples

```
data(AAMetric)
plot(AAMetric[,1], AAMetric[,2], pch = AminoAcids)

cor(AAMetric, AAMetric.Atchley)
```

AAMetric.Atchley *Amino Acid Metric Solution (Atchley et al 2005)*

Description

Atchley et al 2005 performed factor analysis on a set of Amino Acid Indices (AA54) and inferred a 5 factor latent variable structure relating amino acid characteristics using SAS. Based on the relationship between factors and variable descriptions, the latent variables are defined as Factor1 (PAH): Polarity, Accessibility, Hydrophobicity; Factor2 (PSS): Propensity for Secondary Structure; Factor3 (MS) : Molecular Size; Factor4 (CC): Codon Composition; Factor5 (EC): Electrostatic Charge. AAMetric.Atchley are scores from the factor analysis which convey the similarities and differences among amino acids (rows) for each latent variable (columns).

Format

Rows are alphabetized Amino Acids and the 5 columns are factors where Factor1 (PAH): Polarity, Accessibility, Hydrophobicity; Factor2 (PSS): Propensity for Secondary Structure; Factor3 (MS) : Molecular Size; Factor4 (CC): Codon Composition; Factor5 (EC): Electrostatic Charge.

Details

54 Amino Acid Indices were selected from www.genome.jp/aaindex to quantify physiochemical attributes. Using Factor Analysis on 5 factors, interpretable latent variables were determined to quantify Amino Acid attributes. These are the scores from the published factor analysis calculated by SAS. The proportion of common variation for each factor are 42.3

Source

Atchley, W. R., Zhao, J., Fernandes, A. and Druke, T. 2005. Solving the sequence "metric" problem: Proc. Natl. Acad. Sci. USA 102: 6395-6400.

References

Atchley, W. R. and Fernandes, A. 2005. Sequence signatures and the probabilistic identification of proteins in the Myc-Max-Mad network. Proc. Natl. Acad. Sci. USA 102: 6401-6406.

See Also

[AAMetric](#)

Examples

```
data(AAMetric.Atchley)
plot(AAMetric.Atchley[,1], AAMetric.Atchley[,2], pch = AminoAcids)

cor(AAMetric, AAMetric.Atchley)
```

Description

Amino Acids have several distinct and overlapping physiochemical characteristics. The single letter abbreviation for each amino acid is sorted alphabetically in the character vector `AminoAcids`. `AAbyGroup`, `small`, `polar`, and `hydrophobic` correspond to this order and describe various amino acid attributes.

Atchley et al 1999 categorized the 20 amino acids according to physiochemical attributes to form 8 functional groups. The group names are alphabetized in `AAGroups`, while `AAbyGroup` orders these names to pair with `AminoAcids`. `small`, `polar`, and `hydrophobic` contain the vector position of amino acids that characterize that attribute.

AA Groups: acidic = DE aliphatic = AGILMV aminic = NQ aromatic = FWY basic = HKR cysteine = C hydroxylated = ST proline = P

```
AminoAcids = c("A", "C", "D", "E", "F", "G", "H", "I", "K", "L", "M", "N", "P", "Q", "R", "S", "T",
"V", "W", "Y")
AAbyGroup = c("aliphatic", "cysteine", "acidic", "acidic", "aromatic", "aliphatic",
"basic", "aliphatic", "basic", "aliphatic", "aliphatic", "aminic", "proline", "aminic", "basic", "hydroxylated",
"hydroxylated", "aliphatic", "aromatic", "aromatic")
AAGroups = c("acidic", "aliphatic", "aminic", "aromatic", "basic", "cysteine", "hydroxylated", "proline")
small = c(1,2,3,6,12,13,16,17,18)
polar = c(2,3,4,7,9,12,14,15,16,17,19,20)
hydrophobic = c(1, 2,5,6,7,8,9,10,11,17,18,19,20)
```

Author(s)

Lisa McFerrin

References

Atchley, W.R., Terhalle, W. and Dress, A. (1999) Positional dependence, cliques and predictive motifs in the bHLH protein domain. *J. Mol. Evol.* 48, 501-516

Examples

```
data(AA54)
AA54_PCA = princomp(AA54, covmat = cov.wt(AA54))

Factor54 = factor.pa.ginv(AA54, nfactors = 5, m=3, prerotate=TRUE, rotate="Promax", scores="regression")
Factor54$loadings[order(Factor54$loadings[,1]),]

require(scatterplot3d)
Factor3d = scatterplot3d(Factor54$scores[,1:3], pch = AminoAcids, main="Factor Scores", box = FALSE, grid=FALSE,
Factor3d$plane3d(c(0,0,0), col="grey")
Factor3d$points3d(c(0,0), c(0,0), c(-3,2), lty="solid", type="l" )
Factor3d$points3d(c(0,0), c(-1.5,2), c(0,0), lty="solid", type="l" )
Factor3d$points3d(c(-1.5,2), c(0,0), c(0,0), lty="solid", type="l" )
Factor3d$points3d(Factor54$scores[hydrophobic,1:3], col="blue", cex = 2.7, lwd=1.5)
Factor3d$points3d(Factor54$scores[polar,1:3], col="green", cex = 3.3, lwd=1.5)
```

```
Factor3d$points3d(Factor54$scores[small,1:3], col="orange", cex = 3.9, lwd=1.5)
legend(x=5, y=4.5, legend=c("hydrophobic", "polar", "small"), col=c("blue", "green", "orange"), pch=21, box.lty
```

bHLH288

Alignment of basic Helix Loop Helix (bHLH) domain data

Description

The bHLH domain has been categorized into 5 major classes (Atchley and Fitch 1997). The bHLH288 dataset contains 288 amino acid sequences with samples from each class. While the basic and helix regions have well defined structures consisting of 13 and 15 amino acids respectively, the loop region has variable length. To prevent gaps, the loop was truncated in some proteins so only 51 sites are retained and partitioned into basic (1-13), helix(14-28), loop (29-36), and helix(37-51) regions.

Details

The bHLH domain is present throughout Eukaryotes and acts as a transcriptional regulator. This alignment consists of 51 sites where the first 13 constitute the basic region responsible for DNA binding. Each of the 2 helices are 15 amino acids in length, while the loop is variable. Groups are specified by several factors, including the E-box binding specification and inclusion or lack of other domains. The 5 groups are designated by their E-box specificity and presence of additional domains where Group A binds to CAGCTG E-box motif, Group B binds to CACGTG E-box motif and is most prevalent, Group C has an additional PAS domain, Group D lacks a basic region, and Group E binds to CACG[C/A]G N-box motif.

Source

Atchley, W.R. and Fitch, W. (1997) A natural classification of the basic helix-loop-helix class of transcription factors. Proc. Natl. Acad. Sci. USA 94: 5172-5176.

Atchley, W.R. and Fernandes, A. (2005) Sequence signatures and the probabilistic identification of proteins in the Myc-Max-Mad network. Proc. Natl. Acad. Sci. USE 102: 6401-6406

Examples

```
data(bHLH288)

#Separate grouping and sequences
grouping = t(bHLH288[,1])
bHLH_Seq = as.vector(bHLH288[,2])
```


factor.pa.ginv

Principal Axis Factor Analysis when $D \gg N$ **Description**

For data with more variables than observations ($D \gg N$), the covariance matrix is singular and a general inverse is used to determine the inverse correlation matrix and estimate scores. Using the principal axes method of Factor Analysis, communalities are estimated by iteratively updating the diagonal of the correlation matrix and solving the eigenvector decomposition. Communalities for each variable are estimated according to the number of factors and convergence is defined by the stabilization of total communalities between iterations.

Usage

```
factor.pa.ginv(r, nfactors = 1, residuals = FALSE, prerotate = FALSE, rotate = "varimax", m = 4, n.obs
```

Arguments

r	Covariance matrix or raw data matrix. A correlation matrix is computed using pairwise deletion.
nfactors	Number of factors to extract. Default is 1.
residuals	logical. If residual matrix is included in result
prerotate	logical. Rotate the loadings using a varimax orthogonal rotation before applying a different rotation.
rotate	"none", "varimax", "promax" rotation applied to the loadings
m	integer. power of the fitting function in a promax rotation. Default is 4.
n.obs	Number of observations used to find the correlation matrix if using a correlation matrix. Used for finding the goodness of fit statistics.
scores	If TRUE, estimate factor scores. If $D \gg N$, ginv(r) is used during the calculation.
force	if TRUE, a square matrix r will be interpreted as a data matrix. The default is FALSE, and square matrices are assumed to represent covariance
SMC	Use squared multiple correlations (SMC=TRUE) or use 1 as initial communality estimate. Try using 1 if imaginary eigen values are reported.
missing	If scores are TRUE, and missing=TRUE, then impute missing values using either the median or the mean
impute	"median" or "mean" values are used to replace missing values
min.err	Iterate until the change in communalities is less than min.err. Default is 0.001
digits	Number of digits to display in output
max.iter	Maximum number of iterations for convergence
symmetric	symmetric=TRUE forces symmetry by just looking at the lower off diagonal values
warnings	warnings=TRUE displays warning messages encountered during estimation

Value

values	Eigen values of the final solution
communality	Communality estimates for each item. These are merely the sum of squared factor loadings for that item.
rotation	which rotation was requested?
n.obs	number of observations specified or found
loadings	An item by factor loading matrix of class "loadings" Suitable for use in other programs (e.g., GPA rotation or factor2cluster.
fit	How well does the factor model reproduce the correlation matrix. (See VSS, ICLUST, and principal for this fit statistic.
fit.off	how well are the off diagonal elements reproduced?
dof	Degrees of Freedom for this model. This is the number of observed correlations minus the number of independent parameters. Let n=Number of items, nf = number of factors then $dof = n * (n-1)/2 - n * nf + nf*(nf-1)/2$
objective	value of the function that is minimized by maximum likelihood procedures. This is reported for comparison purposes and as a way to estimate chi square goodness of fit. The objective function is $\log(\text{trace}((FF' + U2)^{-1} R)) - \log(\text{trace}((FF' + U2)^{-1} R)) - n.\text{items}$.
STATISTIC	If the number of observations is specified or found, this is a chi square based upon the objective function, f. Using the formula from factanal(which seems to be Bartlett's test) : $\chi^2 = (n.\text{obs} - 1 - (2 * p + 5)/6 - (2 * \text{factors})/3) * f$
PVAL	If n.obs > 0, then what is the probability of observing a chisquare this large or larger?
Phi	If oblique rotations (using oblimin from the GPArotation package or promax) are requested, what is the interfactor correlation.
communality.iterations	The history of the communality estimates. Probably only useful for teaching what happens in the process of iterative fitting.
residual	If residuals are requested, this is the matrix of residual correlations after the factor model is applied.

Note

This is a direct adaptation from the factor.pa function implemented in the psych package.

Author(s)

Lisa McFerrin

References

Gorsuch, Richard, (1983) Factor Analysis. Lawrence Erlbaum Associates. Revelle, William. (in prep) An introduction to psychometric theory with applications in R. Springer. Working draft available at <http://personality-project.org/r/book.html>

See Also[Promax.only](#)**Examples**

```
#compare Principal Components and Factor Analysis methods on Amino Acid data with D>>N

data(AA54)
AA54_PCA = princomp(AA54, covmat = cov.wt(AA54))

Factor54 = factor.pa.ginv(AA54, nfactors = 5, m=3, prerotate=TRUE, rotate="Promax", scores="regression")
Factor54$loadings[order(Factor54$loadings[,1]),]

require(scatterplot3d)
Factor3d =scatterplot3d(Factor54$scores[,1:3], pch = AminoAcids, main="Factor Scores", box = FALSE, grid=FALSE,
Factor3d$plane3d(c(0,0,0), col="grey")
Factor3d$points3d(c(0,0), c(0,0), c(-3,2), lty="solid", type="l" )
Factor3d$points3d(c(0,0), c(-1.5,2), c(0,0), lty="solid", type="l" )
Factor3d$points3d(c(-1.5,2), c(0,0), c(0,0), lty="solid", type="l" )
Factor3d$points3d(Factor54$scores[hydrophobic,1:3], col="blue", cex = 2.7, lwd=1.5)
Factor3d$points3d(Factor54$scores[polar,1:3], col="green", cex = 3.3, lwd=1.5)
Factor3d$points3d(Factor54$scores[small,1:3], col="orange", cex = 3.9, lwd=1.5)
legend(x=5, y=4.5, legend=c("hydrophobic", "polar", "small"), col=c("blue", "green", "orange"), pch=21, box.lty

cor(AA54_PCA$scores, Factor54$scores)
```

FactorTransform

*Metric Solution for Amino Acid characters***Description**

Based off the work done by Atchley et al 2005, Amino Acids are transformed into 5 metrics according to factor analysis scores representing Factor1 (PAH): Polarity, Accessibility, Hydrophobicity; Factor2 (PSS): Propensity for Secondary Structure; Factor3 (MS) : Molecular Size; Factor4 (CC): Codon Composition; Factor5 (EC): Electrostatic Charge. These numerics provide a biologically meaningful value that establishes a platform capable of handling rigorous statistical techniques such as analysis of variance, regression, discriminant analysis, etc.

Usage

```
FactorTransform(Source, Search = AminoAcids, Replace = AAMetric.Atchley, Factor = 1, bycol = TRUE, Sec
```

Arguments

Source	Vector, Matrix or List of Amino Acid Sequences using the single character abbreviation~
Search	Vector of symbols to search over. Default is the list of Amino Acids.

Replace	Vector or Matrix of values to replace Search items. Rows of Replace correspond to elements of Search when byCol = TRUE.
Factor	If Replace is a matrix, Factor designates which vector of Replace is used.
bycol	logical. Designates if Replace is oriented so that columns correspond to replaceable elements
SeqName	Vector of sequence names
alignment	if FALSE, result is a list. If TRUE result is a matrix and hanging rows are filled with fillblank
fillblank	if alignment is TRUE, trailing sites are filled with this value. Default is NA, but can be numeric.

Value

A list or matrix containing numeric representations of the sequences is returned. If alignment is FALSE, each sequence is a new element in the list containing a vector of values with length corresponding to the length of the original sequence. If alignment is TRUE, a matrix is returned with each row representing a sequence metric. If the sequence lengths were unequal, trailing blanks are specified by the fillblank parameter.

Author(s)

Lisa McFerrin

References

Atchley, W. R., Zhao, J., Fernandes, A. and Drueke, T. 2005. Solving the sequence "metric" problem: Proc. Natl. Acad. Sci. USA 102: 6395-6400.

See Also

[lapply](#), [replace](#)

Examples

```
FactorTransform("HDMD", Replace= AAMetric.Atchley)

data(bHLH288)
bHLH_Seq = as.vector(bHLH288[,2])
bHLH_ccList = FactorTransform(bHLH_Seq, Factor=4)
bHLH_ms     = FactorTransform(bHLH_Seq, Factor=3, alignment=TRUE)

bHLH_ms[c(20:25, 137:147, 190:196, 220:229, 264:273),1:8]
```

Loadings.variation *Proportional and Cumulative Variation of Loading Components*

Description

Principal Component Analysis (PCA) methods `prcomp` and `princomp` do not accurately reflect the proportion of total variation of each principal component. Instead `princomp` calculates these values on the eigenvalue adjusted data, which misleadingly indicates that each component contributes equally to the variability in the loadings output. `prcomp` does not report the proportion of variability. To rectify this, `Loadings.variation` displays the relative and cumulative contribution of variation for each component by accounting for all variability in data. Component variation is reported by the lambda value (which corresponds to the eigenvalue in `princomp`), while the proportion and cumulative variation relate these values to the total variability in data.

Usage

```
Loadings.variation(sdev, digits = 5)
```

Arguments

<code>sdev</code>	vector of standard deviations for each component
<code>digits</code>	number of decimal places to retain. Default is 5.

Details

For each component:

$\text{Lambda} = \text{sdev}^2$ Component Variance
 $\text{PTV} = \text{Lambda} / \text{sum}(\text{Lambda})$ Proportion of Total Variation
 $\text{CTV} = \text{cumsum}(\text{PTV})$ Cumulative Total Variation

All variability is accounted for in Principal Components, where each component is orthogonal and in decreasing order of variation explained. This allows PTV to be calculated as a proportion of the sum of individual variances and $\text{CTV}=1$ when accounting for all components.

Value

labeled matrix of variation for loading components. Lambda represents the variation for each component, PTV is the Proportion of Total Variation and CTV is the Cumulative Proportion of Total Variation. Values are rounded according to the number of digits specified.

Author(s)

Lisa McFerrin

See Also

`prcomp`, `princomp`

Examples

```
PCA_SVD = prcomp(USArrests, scale = TRUE)
PCA_SVD$rotation
Loadings.variation(PCA_SVD$sdev)
```

```
PCA_EIG = princomp(USArrests, cor = TRUE)
PCA_EIG$loadings
Loadings.variation(PCA_EIG$sdev)
```

MolecularEntropy

Molecular Entropy for DNA or Amino Acid Sequences

Description

Entropy (H) is a measure of uncertainty for a discrete random variable and is analogous to variation in continuous data. Traditionally the logarithm base for entropy is calculated with unit bits (b=2), nats (b=e) or dits (b=10). Alternatively, entropy estimates can be normalized to a common scale where $0 \leq H \leq 1$ by setting $b=n$, the number of possible states. For DNA (n=4 nucleotide) or protein (n=20 amino acid) sequences, normalized entropy H=0 indicates an invariable site while H=1 represents a site where all states occur with equal probability.

Atchley et al 1999 categorized amino acids according to physiochemical attributes to form (n=8) functional groups. In conjunction with the AA entropy, the GroupAA entropy value may provide insight to differences in functional and phylogenetic variation.

AA Groups: acidic = DE aliphatic = AGILMV aminic = NQ aromatic = FWY basic = HKR cysteine = C hydroxylated = ST proline = P

Gaps are ignored on a site by site basis so the entropy values may have different number of observations among sites. Sequences must be of the same length.

Usage

```
MolecularEntropy(x, type)
```

Arguments

x	matrix, vector, or list of aligned DNA or Amino Acid sequences. If matrix, rows must be sequences and columns individual characters of the alignment. vector and list structures will be coerced into this format.
type	"DNA", "AA", or "GroupAA" method for calculating and normalizing the entropy value for each column (site)

Value

counts	matrix of integers counting the presence of each character (DNA, AA, or GroupAA) at each site
freq	matrix of character (DNA, AA, or GroupAA) frequencies. These are simply character counts divided by total number of (non-gap) characters at each site
H	vector of Entropy values for each site

Author(s)

Lisa McFerrin

References

- Atchley, W.R., Terhalle, W. and Dress, A. (1999) Positional dependence, cliques and predictive motifs in the bHLH protein domain. *J. Mol. Evol.* 48, 501-516
- Kullback S. (1959) *Information theory and statistics*. Wiley, New York

Examples

```
data(bHLH288)
bHLH_Seq = bHLH288[,2]
MolecularEntropy(bHLH_Seq, "AA")
MolecularEntropy(bHLH_Seq, "GroupAA")
```

MolecularMI

Molecular Mutual Information

Description

Mutual information (MI) represents the interdependence of two discrete random variables. Thus MI quantifies the reduction in uncertainty of one variable given the knowledge of a second variable. Placing entropy values on the diagonal of a MI matrix forms a structure comparable to a covariance matrix appropriate for variability decomposition. MI identifies pairs of statistically dependent or coupled sites where MI=1 indicates complete coupling.

Usage

```
MolecularMI(x, type, normalized)
```

Arguments

x	matrix, vector, or list of aligned DNA or Amino Acid sequences. If matrix, rows must be sequences and columns individual characters of the alignment. vector and list structures will be coerced into this format.
type	"DNA", "AA", or "GroupAA" method for calculating and normalizing the entropy value for each column (site)

normalized method of normalization. If "NULL" or not provided, $MI[i,j] = H(x[i]) + H(x[j]) - H(x[i],x[j])$ for $i,j=1..n$ where n is the number of sites. Otherwise, MI is normalized by some leveling constant. see [NMI](#)

Value

$n \times n$ matrix of mutual information values (DNA, AA, GroupAA), where n is the number of sites in the alignment. The diagonal contains the entropy values for that site.

Author(s)

Lisa McFerrin

See Also

[MolecularEntropy](#), [NMI](#)

Examples

```
data(bHLH288)
bHLH_Seq = bHLH288[,2]
bHLH.MIAA = MolecularMI(bHLH_Seq, "AA")
bHLH.MIFG = MolecularMI(bHLH_Seq, "GroupAA")

##Compare Entropy values
MolecularEntropy(bHLH_Seq, "AA")$H
diag(bHLH.MIAA)
diag(bHLH.MIFG)

plot(diag(bHLH.MIFG), type = "h", ylab="Functional Entropy", xlab="site")
```

NMI

Normalized Mutual Information

Description

Mutual information (MI) represents the interdependence of two discrete random variables and is analogous to covariation in continuous data. The intersection of entropy space of two random variables bound MI and quantifies the reduction in uncertainty of one variable given the knowledge of a second variable. However, MI must be normalized by a leveling ratio to account for the background distribution arising from the stochastic pairing of independent, random sites. Martin et al. (2005) found that the background MI, particularly from phylogenetic covariation, has a contributable effect for multiple sequence alignments (MSAs) with less than 125 to 150 sequences.

NMI provides several methods for normalizing mutual information given the individual and joint entropies.

Usage

NMI(Hx, Hy, Hxy, type = c("NULL", "marginal", "joint", "min.marginal", "max.marginal", "min.condition

Arguments

Hx	Marginal entropy for a discrete random variable (x)
Hy	Marginal entropy for a discrete random variable (y)
Hxy	Joint entropy for a discrete random variables (x and y)
type	method of normalization. Default is "NULL" and the Mutual Information is calculated as $MI = Hx+Hy-Hxy$. Other methods include "marginal", "joint", "min.marginal", "max.marginal", "min.conditional", "max.conditional". See details below.

Details

If any denominator is zero, $MI=0$. Otherwise

Methods of Normalization:

marginal $MI = 2*(Hx + Hy - Hxy) / (Hx + Hy)$ joint $MI = 2*(Hx + Hy - Hxy) / (Hxy)$
 min.marginal $MI = (Hx + Hy - Hxy) / \min(Hx,Hy)$ max.marginal $MI = (Hx + Hy - Hxy) / \max(Hx,Hy)$
 min.conditional $MI = (Hx + Hy - Hxy) / \min(Hx.y,Hy.x)$ max.conditional $MI = (Hx + Hy - Hxy) / \max(Hx.y,Hy.x)$

Value

normalized mutual information value

Author(s)

Lisa McFerrin

References

Martin, L.C., G. B. Gloor, et al. (2005). Using information theory to search for co-evolving residues in proteins. *Bioinformatics*. 21, 4116-24.

See Also

[MolecularEntropy](#), [MolecularMI](#),

pairwise.mahalanobis *Mahalanobis distances for grouped data*

Description

Returns a square matrix of Mahalanobis distances by doing a pairwise comparison of group means using the correlation between variables.

Usage

```
pairwise.mahalanobis(x, grouping = NULL, cov = NULL, inverted = FALSE, digits = 5, ...)
```

Arguments

x	vector or matrix of data with N observations and D variables. If grouping is not specified, the first column is used for grouping observations.
grouping	vector of characters or values designating group classification for observations.
cov	Covariance matrix (DxD) of the distribution
inverted	logical. If TRUE, cov is the inverse of the covariance matrix.
digits	number of decimals to keep for the means, cov and distance values
...	passed to mahalanobis for computing the inverse of the covariance matrix (if inverted is false).

Details

To determine the distance between group i and group j, the difference of group means for each variable are compared. For a (NxD) data matrix with m groups, a matrix of mxD means and a correlation matrix of DxD values are calculated. pairwise.mahalanobis calculates the mahalanobis distance for all possible group combinations and results in a mxm square distance matrix with m choose 2 distinct pairwise measures.

Value

means	(mxD) matrix of group means for each variable
cov	(DxD) covariance matrix of centered and scaled data, so it's actually the correlation matrix
distance	(mxm) matrix of squared mahalanobis distances

Author(s)

Lisa McFerrin

See Also

[mahalanobis](#)

Examples

```

data(bHLH288)
grouping = t(bHLH288[,1])
bHLH_Seq = as.vector(bHLH288[,2])
bHLH_pah = FactorTransform(bHLH_Seq, alignment=TRUE)

Mahala1 = pairwise.mahalanobis(bHLH_pah, grouping, digits = 3)
D = sqrt(Mahala1$distance)
D

```

Promax.only

Promax rotation (without prior Varimax rotation)

Description

Promax.only is an oblique rotation of factor loadings. This function is directly derived from the Promax function in the psych package, but only performs the promax rotation without first specifying a varimax orthogonal rotation. Further specifying the power of the fitting function allows for greater versatility.

Usage

```
Promax.only(x, m = 4, rotate.structure=NULL)
```

Arguments

x	matrix of factor loadings
m	power of fitting function
rotate.structure	rotation matrix if loadings have been prerotated. Default is the identity matrix.

Details

An oblique factor rotation will rescale the loadings with factors having correlated structure Phi

Value

loadings	Oblique factor loadings
rotmat	Rotation matrix structure. If rotated.structure supplied, it will be factored into rotmat.
Phi	Correlation matrix structure of Factors

Note

Adapted directly from Promax of the psych package

Author(s)

Lisa McFerrin

References

Hendrickson, A. E. and White, P. O, 1964, British Journal of Statistical Psychology, 17, 65-70.

See Also

[promax, factor.pa.ginv](#)

Examples

```
##compare to promax and Promax solutions  
fa <- factanal( ~., 2, data = swiss)  
Promax(loadings(fa))  
Promax.only(loadings(fa))
```

Index

*Topic **datasets**

AA54, [4](#)

AAMetric, [5](#)

AAMetric.Atchley, [6](#)

bHLH288, [8](#)

*Topic **package**

HDMD-package, [3](#)

AA54, [4](#)

AAbyGroup (AminoAcids), [7](#)

AAGroups (AminoAcids), [7](#)

AAMetric, [5](#), [6](#)

AAMetric.Atchley, [5](#), [6](#)

AminoAcids, [7](#)

bHLH288, [8](#)

factor.pa.ginv, [5](#), [9](#), [20](#)

FactorTransform, [11](#)

HDMD (HDMD-package), [3](#)

HDMD-package, [3](#)

hydrophobic (AminoAcids), [7](#)

lapply, [12](#)

Loadings.variation, [13](#), [13](#)

mahalanobis, [18](#)

MolecularEntropy, [14](#), [16](#), [17](#)

MolecularMI, [15](#), [17](#)

NMI, [16](#), [16](#)

pairwise.mahalanobis, [18](#)

polar (AminoAcids), [7](#)

prcomp, [13](#)

princomp, [13](#)

promax, [20](#)

Promax.only, [11](#), [19](#)

psych, [3](#)

replace, [12](#)

small (AminoAcids), [7](#)